Práctico 1

En este práctico trabajaremos con el conjuto de datos de movie reviews que utilizaron en la materia *Aprendizaje Supervisado*. La tarea es clasificar en positiva o negativa cada una de las revisiones de los usuario.

Durante esta etapa implementaremos modelos MLP básicos, y veremos los diferentes hiperparámetros y arquitecturas que podemos elegir. Compararemos además dos tipos de representaciones comunes para en el procesamiento del lenguaje natural: utilizando TfIDF y utilizando embeddings de palabras.

Para resolver ambos ejercicios, les proveemos un esqueleto que pueden completar.

Ejercicio 1

1. Procesar el conjunto de datos para obtener una representación TfIDf de cada review.
    - Hint: User el código de *Aprendizaje Supervisado*
    - Hint: El método de fit de Keras crea internamente un conjunto de datos de validación, por lo que no tienen que preocuparse por eso.
2. Construir un pipeline de clasificación con un modelo Keras MLP.
3. Entrenar uno o varios modelos (con dos o tres es suficiente, veremos más de esto en el práctico 2). Evaluar los modelos en el conjunto de test.
4. Reportar los hyperparámetros y resultados de todos los modelos entrenados. Para esto, pueden utilizar una notebook o un archivo (pdf|md). Dentro de este reporte tiene que describir:
    - Hyperparámetros con los que procesaron el dataset: tamaño del vocabulario, normalizaciones, etc.
    - Las decisiones tomadas al construir cada modelo: regularización, dropout, número y tamaño de las capas, optimizador.
    - Proceso de entrenamiento: división del train/test, tamaño del batch, número de épocas, métricas de evaluación. Seleccione los mejores hiperparámetros en función de su rendimiento. El proceso de entrenamiento debería ser el mismo para todos los modelos.
    - (Punto estrella) Analizar si el clasificador está haciendo overfitting. Esto se puede determinar a partir del resultado del método fit.

Ejercicio 2

1. Procesar el conjunto de datos para obtener una representación basada en embeddings de palabras de cada revisión. Pueden utilizar cualquier wordvector, pero les recomendamos FastText.

Estamos utilizando una versión filtrada de FastText, donde sólo incluímos las palabras en el vocabulario del dataset de movies. Para más información, ver el script `filter_fasttext.py` .

1. Repetir los pasos 2 a 4 del ejercicio 1